

Amendments to the Claims:

The following listing of claims replaces all prior versions, and listings, of claims in the present application.

Listing of the Claims:

1. (currently amended) A method of communicating datagrams between ~~terminals~~ a first terminal having a first address, and a second terminal having a second address, of a communication system, wherein each datagram comprises redundancy check data used to verify datagram integrity, the method comprising the steps of:

generating a first datagram comprising message data and first redundancy check data, which first redundancy check data is computed in dependence on the message data and the first address;

sending the first datagram from a the first terminal to a the second terminal;

verifying the integrity of the first datagram;

generating a second datagram which comprises second redundancy check data, which second redundancy check data is computed in dependence on response data and the first redundancy check data;

sending the second datagram from the second terminal to the first terminal;

verifying the integrity of the second datagram; and,

in the case where the integrity of the second datagram is confirmed, identifying that the second datagram is the response to the first datagram.

2. (original) A method according to claim 1 wherein the step of verifying the integrity of the first datagram comprises the steps of:

calculating third redundancy check data in dependence on the message data; comparing third redundancy check data with first redundancy check data; and

determining the integrity of the first datagram in dependence on the comparison.

3. (original) A method according to claim 1 wherein the step of verifying the integrity of the second datagram comprises the steps of:

calculating fourth redundancy check data in dependence on the response data and first redundancy check data;

comparing fourth redundancy check data with second redundancy check data; and

determining the integrity of the second datagram in dependence on the comparison.

4. (original) A method according to claim 1 wherein computing second redundancy check data comprises the steps of:

initialising a first redundancy check data generator in dependence on the first redundancy check data; applying response data to the redundancy check data generator; and

determining second redundancy check data in dependence on the value of the first redundancy check data generator.

5. (previously presented) A method according to claim 1, wherein the step of verifying the integrity of the second datagram comprises the steps of:

initialising a second redundancy check data generator in dependence on the first redundancy check data;

applying response data of the second datagram and second redundancy check data to the generator, which response data was that used to compute the second redundancy check data; and

determining the integrity of the second datagram in dependence on the value of the second redundancy check data generator.

6. (previously presented) A method according claim 1, wherein the second datagram further comprises the response data.

7. (previously presented) A communications system comprising a plurality of terminals, wherein each terminal employs the method of claim 1.

8. (canceled)

9. (previously presented) A terminal for use in the communications system according to claim 7, the terminal comprising:

a first port operable to receive a datagram from another terminal; a processor operable to:

decode a received datagram;

compute redundancy check data;

compare redundancy check data; and

generate a datagram;

a first store operable to store program code instructions;

a second store operable to store redundancy check data; a second port operable to send datagrams to another terminal; and

a third port operable to exchange data with a host application.

10. (original) A terminal according to claim 9, in which the first store is non-volatile.

11. (original) A terminal according to claim 9, further comprising a redundancy check data generator.

12. (canceled)

13. (canceled)